

An efficient intrusion detection systems in fog computing using forward selection and BiLSTM

Fadi Abu Zwayed¹, Mohammed Anbar¹, Selvakumar Manickam¹, Yousef Sanjalawe², Hamza Alrababah³, Iznan H. Hasbullah¹, Noor Almi'ani¹

¹National Advanced IPv6 Centre of Excellence (NAv6), University Sains Malaysia, Penang, Malaysia

²Department of Cybersecurity, Faculty of Information Technology, American University of Madaba, Amman, Jordan

³School of Computing, Skyline University College, University City of Sharjah, Sharjah, United Arab Emirates

Article Info

Article history:

Received Jul 5, 2023

Revised Dec 11, 2023

Accepted Dec 20, 2023

Keywords:

Bidirectional long short-term
Cybersecurity
Deep learning
Feature selection
Fog computing
Internet of things
Intrusion detection system

ABSTRACT

Intrusion detection systems (IDS) play a pivotal role in network security and anomaly detection and are significantly impacted by the feature selection (FS) process. As a significant task in machine learning and data analysis, FS is directed toward pinpointing a subset of pertinent features that primarily influence the target variable. This paper proposes an innovative approach to FS, leveraging the forward selection search algorithm with hybrid objective/fitness functions such as correlation, entropy, and variance. The approach is evaluated using the BoT-IoT and TON_IoT datasets. By employing the proposed methodology, our bidirectional long-short term memory (BiLSTM) model achieved an accuracy of 98.42% on the TON_IoT dataset and 98.7% on the BoT-IoT dataset. This superior classification accuracy underscores the efficacy of the synergized BiLSTM deep learning model and the innovative FS approach. The study accentuates the potency of the proposed hybrid approach in FS for IDS and highlights its substantial contribution to achieving high classification performance in internet of things (IoT) network traffic analysis.

This is an open access article under the [CC BY-SA](#) license.



Corresponding Author:

Mohammed Anbar

National Advanced IPv6 Centre of Excellence (NAv6), University Sains Malaysia

11800 USM, Penang, Malaysia

Email: anbar@usm.my

1. INTRODUCTION

The proliferation of the Internet and its associated technologies has led to an unprecedented surge in data communication among devices. This upsurge underscores the increasing reliance on digital channels and the growing need to ensure the integrity and security of the transmitted information. As a linchpin of modern network security, IDSs are critical in safeguarding data against potential threats and breaches.

Recent years have seen the meteoric rise of IoT devices, revolutionizing everything from home automation to industrial processes. While these devices bring numerous advantages, they also present significant challenges, primarily due to the sheer volume of data they generate. Processing and managing this colossal real-time data necessitates solutions beyond traditional cloud computing paradigms. Enter fog computing: an evolutionary step from cloud computing, which emphasizes decentralized, local data processing, offering real-time insights closer to the source of data generation [1].

However, as with any evolutionary technology, fog computing introduces its own set of challenges. While advantageous for processing, decentralization inadvertently amplifies potential security threats. Given the critical nature of many IoT applications, these threats can have severe repercussions. As such, the role of IDSs becomes even more crucial in the context of fog computing environments [2].

Historically, IDSs have shown limitations when confronted with the high-dimensional nature of network data, especially in scenarios blending IoT with fog computing [3]. Recognizing this, the research community posited feature selection (FS) as a potential solution. By reducing dimensionality and enhancing data interpretability, FS promises to rejuvenate the effectiveness of IDSs [4]. However, the existing FS methodologies, predominantly relying on singular statistical measures, often fail to address the multifaceted challenges of fog environments.

Traditional FS methods that hinge on a single statistical measure might provide insights into specific aspects of the data but often miss out on capturing the holistic nature of the information. For instance, while correlation can offer insights into linear relationships between features, it might not capture intricate non-linear interdependencies that other metrics like entropy or variance can elucidate. Consequently, relying solely on one measure can inadvertently lead to omitting critical features or including redundant ones.

This research aims to bridge this gap by proposing an innovative method for FS that marries the strengths of the forward selection algorithm with hybrid objective functions. These functions are a cocktail of correlation, entropy, and variance metrics designed to select features that balance being informative and ensuring mutual exclusivity. Such a balance is no longer a luxury but a necessity when dealing with high-dimensional IDS data [5], [6].

But the innovation doesn't stop at FS. Recognizing the potential of deep learning, this research further integrates the bidirectional BiLSTM model into the IDS framework. BiLSTM, an advanced avatar of the traditional recurrent neural network (RNN), is renowned for its capability to interpret sequential data, making it an ideal choice for network traffic analysis [7].

In a realm inundated with intrusion detection methodologies, the bespoke approach to feature selection tailored for fog computing environments sets this research apart. While traditional systems have relied on singular metrics or straightforward algorithms, this work pioneers a synergistic approach, combining the robustness of forward selection, intricate statistical measures, and the predictive prowess of BiLSTM. Such a holistic technique, to the best of our knowledge, has not been explored, and especially within the context of fog computing architectures [8].

To encapsulate, the study's vision is twofold: first, to introduce and validate a novel FS methodology tailored for IDSs in fog computing environments, and second, to harness the power of deep learning. Specifically, BiLSTM enhances the detection accuracy of intrusion detection.

The contributions of this research are manifold:

- The introduction of a bespoke FS method for IDSs, meticulously crafted for the challenges of fog computing environments.
- A pioneering approach that synergizes forward selection, intricate statistical measures, and BiLSTM to push the frontiers of intrusion detection.
- A proactive response to the pressing security demands of the IoT era, ensuring robust, resilient, and efficient IDS performance within fog computing architectures.

As the internet security landscape continues to evolve, it's imperative to delve deeper into the intricacies of the methodologies that promise to safeguard its future. The subsequent sections aim to unravel this meticulous research journey. The discussion commences with a detailed exploration of related works, setting the stage for a deeper understanding of the research context. Moving forward, the manuscript delves into the proposed methodology, elucidating the nuances of the hybrid objective function, the forward selection algorithm, and the configuration specifics of the BiLSTM model. Central to this research evaluation is using the Bot-IoT and TON_IoT datasets. These datasets, chosen for their comprehensive representation of diverse network scenarios, are the foundation for assessing the system's efficacy. The results, benchmarked against crucial metrics like accuracy, precision, recall, and F1 score, offer a rigorous evaluation of the system's capabilities, ensuring that the presented contributions are firmly rooted in empirical evidence.

2. BACKGROUND

More details about fog computing, its security flaws, cyberattacks, and IDS will be presented here. Following a brief introduction to fog computing and its significance, various security concerns and malicious attacks that can affect IoT devices and fog are outlined. Finally, the various types of IDS and their significance in addressing these issues will be discussed.

2.1. Fog computing

The hierarchical architecture of fog computing integrates cloud resources with edge devices, and fog nodes see Figure 1. Cloud resources provide storage and global-scale services, while fog nodes process data locally, reducing network congestion and latency [9]. IoT devices collect and generate data at the bottom

layer. This architecture improves data processing efficiency, reduces network traffic, and enables real-time applications in domains like smart cities and healthcare.

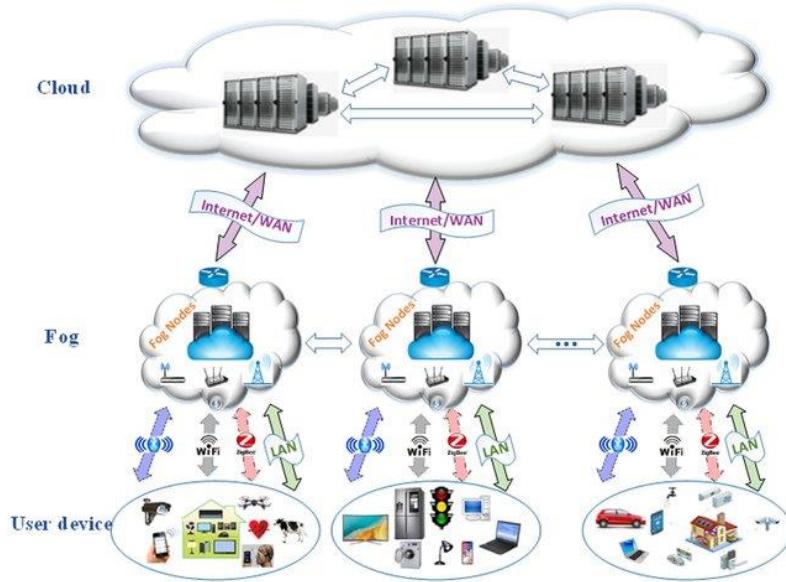


Figure 1. Fog computing architecture [10]

Fog systems, mini-clouds situated at the network's edge near the user, possess scalable memory, processing, and storage capacities [11]. They support real-time applications by reducing latency through fog nodes like access points, switches, servers, controllers, routers, gateways, and storage devices [12]. End users interact with these nodes instead of cloud data centers, which saves energy and enhances response time. Fog nodes can be classified into two types: ones that only produce and sense data and intelligent ones that sense data and perform initial data processing using advanced computational abilities [13]. Fog computing assists in load balancing, reducing latency, and maintaining service quality.

2.2. Fog security issues

The fog platform interposed between users and the cloud engenders numerous potential vulnerabilities, making it susceptible to multiple attacks. Currently, fog computing is employed to improve website performance. They can handle HTTP requests using fog computing, receiving and processing multiple requests simultaneously while managing various user files [14]. This makes fog computing vulnerable to attacks like cross-site scripting (XSS) and injection attacks based on lousy input that has not been checked. Fog nodes participate in mutual communication with many (IoT) devices. This leaves them open to a security threat known as a man-in-the-middle (MITM) attack. In such an attack, a malefactor positioned in the network's core intercepts communications modifies data, and impairs service provision. Figure 1 shows that the fog platform can connect to sensors, laptops, phones, and other IoT devices used by regular people. This connectivity can be either wired or wireless. Given the accessibility of the fog platform, it becomes prone to resource exploitation attacks, such as denial of service (DoS) and distributed denial of service (DDoS) [15]. Such attacks pose significant threats to the IoT infrastructure, negatively impacting its performance and causing substantial harm. For instance, scanning attacks seek to gather data, like available system services and open ports, while backdoor attacks leverage concealed malware to control IoT systems remotely. Techniques for breaching the passwords of IoT devices are prevalent within the fog environment. Furthermore, fog computing is susceptible to ransomware attacks, which inhibit a user's ability to access an IoT device or service [16].

2.3. Intrusion detection system

Fog networks, susceptible to numerous security risks such as traffic interception and malware distribution, employ security measures like firewalls, encryption, and Intrusion IDS for protection [17]. IDSs, utilized in fog nodes, analyze data using machine learning and deep learning to spot threats [18]. They come in two types-network intrusion detection systems (NIDSs) that scan network traffic for harmful activity and

host intrusion detection systems (HIDSs) that monitor hosts for malicious actions [19]. These systems are anomaly-based, detecting unusual behavior, or misuse-based, looking for known attack patterns. They operate in distributed or centralized architectures, with NIDSs embedded in fog nodes for broader network coverage [20]. The development of IDSs in fog computing is an ongoing process due to the emergence of new threats [21].

2.4. Features selection

FS is a critical process in deep learning and data mining, especially with large datasets [22]. It identifies the most essential features, reduces data dimensionality, and boosts model performance by removing unneeded features [23]. FS methods include filter methods, which evaluate feature importance independently [24]; wrapper methods, which use predictive models to score feature subsets; and embedded methods, which perform FS during model training. Recently, hybrid methods combining the strengths of individual methods have emerged. FS is crucial in intrusion detection systems, as it enhances the performance of models by focusing on the most informative features [25]. This improves efficiency, increases accuracy, reduces overfitting, and strengthens the detection and classification of intrusive activities in IoT environments [26], [27].

3. RELATED WORK

The related work section explores the prior research and methodologies that form the foundation for the proposed approach. Various FS techniques will be explored, focusing on the forward selection method. We'll examine the incorporation of hybrid objective or fitness functions in FS and how pairwise measures such as correlation, entropy, and variance play a crucial role in this process.

Improving the accuracy of IDS is the primary focus of the hybrid neural network (HNN) model, as introduced in the study [28]. This model performs better by combining multi-feature correlation analysis and temporal-spatial analysis. Through real-world dataset experiments, it surpasses traditional ML algorithms in accuracy, precision, recall, and F1-score measures. The effectiveness of this model in detecting intrusion activities is amplified by its ability to capture complex feature relationships and analyze temporal-spatial patterns.

Continuing in the same direction, enhancing the efficiency of IDS by employing an FS strategy is the focus of the study [29]. In response to the complexity of voluminous network traffic data, the researchers propose FS and eliminating irrelevant ones to improve the accuracy of classification algorithms. The study introduces three FS and ranking techniques-information gain, gain ratio, and correlation FS to select and rank the top features. The narrowed-down selection of six features from an initial set of 41 is then tested using three classifiers-k-nearest neighbor, naïve bayes, and neural network-based multilayer perceptron. The findings indicate that a high attack classification accuracy can be achieved by combining the best features from different methods.

Simultaneously, a novel method for anomaly detection in fog computing is introduced in the study [30], utilizing genetic algorithm (GA) based FS and naïve bayes classification. Recognizing fog computing's pivotal role in handling massive IoT data and its vulnerability to security threats, the authors employ a GA for selecting features that substantially aid in anomaly detection. This process improves classification efficiency by reducing data dimensionality. The naïve bayes classifier, a probabilistic tool premised on feature independence, is used for classification. The fusion of GA-based FS and naïve bayes improves anomaly detection in fog computing environments, lowers false alarm rates, and boosts detection rates, thus offering an efficient security solution.

A unique approach for FS and extraction in anomaly-based (IDS) within the (IoT) ecosystem is the crux of the study [31]. Addressing the difficulties of exploiting all features due to the diverse characteristics of IoT, the authors utilize two entropy-based methods-information gain (IG) and gain ratio (GR) for FS and extraction. Mathematical set theory is also employed for optimal feature extraction. Training and testing of the model are conducted on the IoT intrusion dataset 2020 (IoTID20) and NSL-KDD dataset using four ML algorithms. The outcome reveals that the method successfully identifies relevant features and outperforms other state-of-the-art studies.

The performance of ML techniques for anomaly detection, with and without FS, is examined in the study [32]. The authors utilize the KDD99 dataset, encompassing many network traffic records. An evaluation is conducted on four ML techniques: decision trees, support vector machines, naive bayes, and k-nearest neighbors, along with two FS techniques: information gain and chi-squared. The findings reveal that FS can enhance the performance of ML techniques for anomaly detection. The decision tree classifier achieves the most significant results with the information gain FS technique. The authors infer from their findings that FS is a promising approach for augmenting the performance of IDSs.

An innovative approach for augmenting the performance of a deep neural network (DNN) based (IDS) is proposed in the study [33]. The authors acknowledge the growing use of DL techniques and their powerful capacity to learn data in depth. Their study focuses on enhancing the DNN-based IDS through a unique FS strategy that employs a fusion of statistical importance using standard deviation and the difference of mean and median. The approach aims to prune features based on their rank, derived from the fusion of statistical importance, to identify relevant features that exhibit high discernibility and deviation, thus aiding in more effective data learning. The performance of the proposed approach is tested using three different intrusion detection datasets: NSL-KDD, UNSW_NB-15, and CIC-IDS-2017. The study reports on performance regarding various evaluation metrics and provides a comparative analysis with existing FS techniques. The results undergo statistical testing using the wilcoxon signed rank test.

Building on this theme of combining traditional and modern techniques for IoT security, [34] crafted an intrusion detection system that harnesses the combined power of machine learning and deep learning. When evaluated on the BoT-IoT dataset, their system showcased an accuracy rate exceeding 99%. Notably, their experimentation revealed the prowess of decision tree and multilayer perceptron models in detecting specific threats like DDoS and DoS attacks. However, a comprehensive comparison with other state-of-the-art methods, especially regarding computational efficiency, was a noticeable gap in their research.

Further pushing the boundaries of innovation in this domain, [35] embarked on a journey to address the challenges of hybrid cloud-fog computing in IoT. Their brainchild, the ConvNeXt-Sf model, is a testament to their ingenuity source. This model is not just another deep learning architecture; it's a reimaged version of ConvNeXt, meticulously transformed to cater to IoT's unique challenges and constraints. When tested on datasets like TON_IoT and BoT-IoT, the results were remarkable. The model's parameters were a mere 1.25% of the original ConvNeXt, and it achieved staggering reductions in training and prediction times by 82.63% and 56.48%, respectively.

4. METHOD

IDS for IoT networks necessitates a precise and reliable framework to safeguard the security and operability of interconnected devices amidst the burgeoning threats in cyberspace. While existing IDS paradigms exhibit robustness, they often encounter computational efficiency and precision challenges, particularly when navigating through high-dimensional data where discerning feature relevance and redundancy are pivotal. These challenges underscore the imperative to explore innovative data classification and feature selection approaches, especially in scenarios inundated with voluminous and diverse data.

To navigate these challenges, this study introduces a nuanced deep-learning framework, emphasizing a bespoke hybrid FS algorithm intricately crafted to sift through dataset features with meticulousness and focus. The hybrid FS algorithm, central to this research, transcends being a mere iterative feature selector. It is a methodological innovation carefully designed to scrutinize and select attributes that are inherently crucial and augment the predictive veracity of subsequent modeling stages. Distinct from conventional methods, which may predominantly rely on a single evaluative metric [36], our algorithm fuses various evaluative lenses, ensuring a comprehensive, balanced, and rigorous feature selection process. This methodological choice, governed by a meticulously tuned fitness score, ensures selections are halted upon reaching a predetermined feature count or when the score descends below a rigorously tested threshold, herein set at 0.2 to ensure the selection of only the most paramount features.

A BiLSTM classifier is meticulously trained and validated upon establishing the selected features. This is not merely enacted as a predictive model but is finessed as a strategic instrument that adapts its parameters to mitigate overfitting and enhance generalization across diverse datasets. The intricate parameter tuning and validation stages, elaborated in subsequent sections, ascertain that the model is robust and adaptable to varied IoT data landscapes.

This research transcends the development of the framework, extending into its rigorous validation. By employing new datasets, TON_IoT and BoT-IoT, the model is subjected to meticulous testing, ensuring its efficacy, reliability, and applicability are validated and contrasted against extant frameworks, thereby substantiating its theoretical robustness and practical applicability, as illustrated in Figure 2.

4.1. Data pre-processing

Data pre-processing is the initial phase in preparing raw data for further stages like feature selection and model training in IDS [37]. It involves transforming raw network traffic data into a meaningful format, helping to eliminate irrelevant or redundant information that could hamper detection performance and increase computational time. The process consists of three main sub-stages: data encoding, handling missing values, and normalization, each of which refines the data for more effective subsequent analysis [38].

4.1.1. Data encoding

Undertaking label encoding is paramount in this study, serving as an integral pre-processing step to transmute categorical data into a numerical format, rendering it amenable for DL models. Each unique category within the dataset is assigned a distinct integer, ensuring a seamless transition from categorical to numerical data. The methodology involves iterating through each column, identifying ‘object’ type data indicative of categorical data, and converting it into an integer representation, ensuring comprehensive data utilization across subsequent analytical stages.

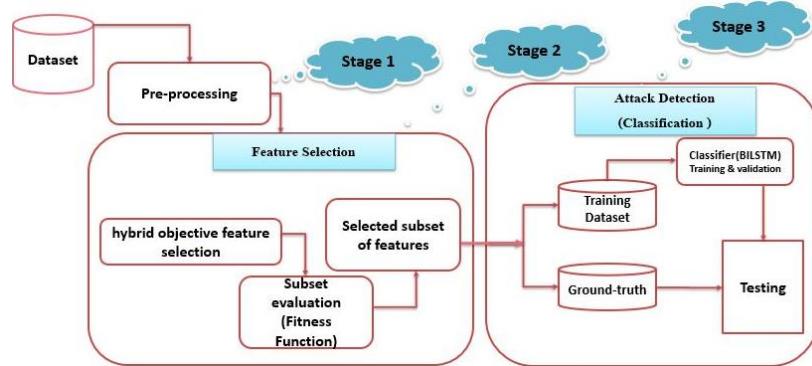


Figure 2. The proposed IDS model architecture

Figure 3 illustrates the encoding process for attacks. Thus, any abnormal traffic, irrespective of the attack form it experiences, is classified under a singular attack category and numerically represented as integer number 1.

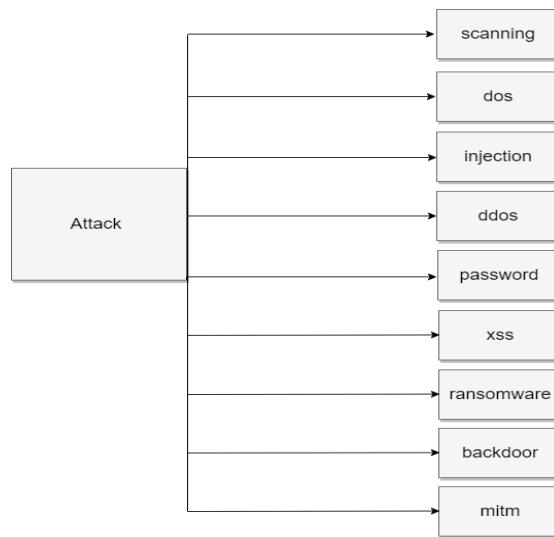


Figure 3. Attacks encoding

4.1.2. Data cleansing and handling missing values

Data cleansing was an imperative phase, crucial for enhancing the quality and reliability of the datasets: TON_IoT and BoT-IoT. This process encompassed several vital steps:

- Handling missing values: absent data can notably impact the subsequent analysis’ accuracy and reliability. In this study, missing values were addressed through mean imputation, a statistical technique wherein missing entries are replaced with the mean value of the non-missing data. The mean was calculated using (1):

$$\text{Mean } (X) = (\Sigma x_i) / n \quad (1)$$

where Σx_i denotes the sum of all observed values for the variable, and n signifies the total number of observed values.

Utilizing mean imputation under the assumption that data was missing completely at random (MCAR) ensured the preservation of the dataset's overall distributional properties and prevented the loss of crucial information [38].

- b. Detecting and removing duplicates: to maintain our analysis's integrity and unbiased nature, duplicate entries within the datasets were identified and eliminated. This prevented the over-representation of specific instances.
- c. Identifying and managing outliers: outliers were detected and managed using the interquartile range (IQR) method. Outliers are points that significantly differ from the rest of the data, which can skew the analysis and subsequent results. The steps for calculating IQR and identifying outliers were:
 - Calculate the IQR as $Q3 - Q1$, where $Q3$ is the third quartile, and $Q1$ is the first quartile.
 - Any data point that fell below $Q1 - 1.5 * IQR$ or above $Q3 + 1.5 * IQR$ was considered an outlier.

Following these steps in the data cleansing stage ensured that the datasets feeding into the subsequent stages of transformation and normalization were reliable, consistent, and of high quality [39].

4.1.3. Data normalization

Data pre-processing often includes normalization to ensure all features are on a consistent scale. One standard method for achieving this is Min-Max scaling. This process transforms each feature in the dataset by subtracting the minimum value of the feature and then dividing it by the difference between the maximum and minimum values of that feature. This transformation ensures that the resulting values lie within the range of $[0, 1]$, as shown by (2):

$$Z = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \quad (2)$$

The feature value is denoted by x , while the value after normalization is represented by Z . The maximum and minimum values of the feature are denoted by x_{\max} and x_{\min} , respectively.

The datasets, namely TON_IoT and BoT-IoT, had been successfully encoded, cleansed, and normalized, as evidenced by the snapshots (Figures 4 and 5) illustrating the state of the data post-preprocessing.

src_ip	src_port	dst_ip	dst_port	proto	service	src_bytes	conn_stm	missed	src_pkts	src_ip_b	dst_pkts	dst_ip_b	dns_QUE	dns_QCL	dns_QTP	dns_rcod	dns_AA	dns_RD	dns_RA	dns_rej	ssl_versi	ssl_ciph	ssl_resu	ssl_ssl	est_ssl	ssl_sub
0.53351	0.72115	0.30329	0.23806	1	0	0	0.5	0	0.05556	0.0163	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	C
0.87993	0.02867	0.7598	0.7902	0.5	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	C
0.87993	0.52332	0.7598	0.16026	0.5	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	C
0.87993	0.7112	1	0.00081	1	0.33333	0	1	0	0	0	0.13333	0.02191	1	0	0	0	0	0	0	0	0	0	0	0	0	C
0.53351	0.50765	0.30329	0.23806	1	0	0	0.5	0	0.05556	0.0163	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	C
0.87993	0.52332	0.7598	0.16026	0.5	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	C
0.87993	0.02867	0.7598	0.7902	0.5	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	C
0.87993	0.02867	0.7598	0.7902	0.5	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	C
0.87993	0.52332	0.7598	0.7902	0.5	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	C
0.87993	0.52332	0.7598	0.16026	0.5	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	C
0.53351	0.57219	0.30329	0.23806	1	0	0	0.5	0	0.05556	0.0163	0	0	1	0	0	0	0	0	0	0	0	0	0	0	C	
0.87993	0.52332	0.7598	0.16026	0.5	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	C
0.87993	0.02867	0.7598	0.7902	0.5	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	C
0.53351	0.60329	0.30329	0.23806	1	0	0	0.5	0	0.05556	0.0163	0	0	1	0	0	0	0	0	0	0	0	0	0	0	C	
0.87993	0.79458	1	0.00081	1	0.33333	0	1	0	0	0	0.13333	0.02191	1	0	0	0	0	0	0	0	0	0	0	0	0	C
0.87993	0.52332	0.7598	0.16026	0.5	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	C
0.87993	0.02867	0.7598	0.7902	0.5	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	C
0.53351	0.87588	0.30329	0.23806	1	0	0	0.5	0	0.05556	0.0163	0	0	1	0	0	0	0	0	0	0	0	0	0	0	C	
0.87993	0.02867	0.7598	0.7902	0.5	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	C
0.87993	0.52332	0.7598	0.16026	0.5	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	C
0.53351	0.88778	0.30329	0.23806	1	0	0	0.5	0	0.05556	0.0163	0	0	1	0	0	0	0	0	0	0	0	0	0	0	C	
0.87993	0.52332	0.7598	0.16026	0.5	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	C	
0.87993	0.02867	0.7598	0.7902	0.5	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	C	
0.87993	0.55528	2.50E-05	0.00676	0.5	0	0	1	0	0.05556	0.01346	0.13333	0.00644	1	0	0	0	0	0	0	0	0	0	0	0	C	
0.87993	0.72742	1	0.00081	1	0.33333	0	1	0	0	0.13333	0.02191	1	0	0	0	0	0	0	0	0	0	0	0	0	C	
0.87993	0.52332	0.7598	0.16026	0.5	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	C	
0.53351	0.9161	0.30329	0.23806	1	0	0	0.5	0	0.05556	0.0163	0	0	1	0	0	0	0	0	0	0	0	0	0	0	C	
0.87993	0.00001	0.7598	0.7902	0.5	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	C	

Figure 4. Snapshot of TON_IoT dataset after pre-processing stage

4.2. The proposed hybrid objectives feature selection method for intrusion detection systems

This study introduces a specialized two-stage FS methodology for identifying crucial attributes in large datasets, particularly for IDS. This hybrid objective FS method integrates various statistical and informational criteria, offering a comprehensive and holistic approach to feature assessment and selection, diverging from traditional FS methods that may rely on a single metric. Subsequent sections will delve into each methodology component, highlighting its innovative contributions and advantages over existing FS methodologies in the IDS domain while providing a transparent and replicable guide for its application.

pkSeqID	stime	figs	figs_num	proto	proto_num	saddr	sport	ddaddr	dport	pkts	bytes	state	state_num	ltime	seq	dur	mean	stddev	sum	min	max	spkts
0.00000	0.57826	0	0.63093	0.19872	0.19262	0.51139	0.57805	0.9257	0.33869	0.88966	0.71311	0.33665	0.60979	0.98915	0.21282	0	0.22231	0	0.05107	0.05107	0	0.48
0.00002	0.57826	0	0.63093	0.19872	0.19262	0.51139	0.57805	0.9257	0.33869	0.88966	0.71311	0.33665	0.60979	0.98915	0.21282	0	0.22231	0	0.05107	0.05107	0	0.48
0.00003	0.57826	0	0.63093	0.19872	0.19262	0.51139	0.57805	0.9257	0.33869	0.88966	0.71311	0.33665	0.60979	0.98915	0.21282	0	0.22231	0	0.05107	0.05107	0	0.48
0.00005	0.57826	0	0.63093	0.19872	0.19262	0.51139	0.57805	0.9257	0.33869	0.88966	0.71311	0.33665	0.60979	0.98915	0.21282	0	0.22231	0	0.05107	0.05107	0	0.48
0.00007	0.57826	0	0.63093	0.19872	0.19262	0.51139	0.57805	0.9257	0.33869	0.88966	0.71311	0.33665	0.60979	0.98915	0.21282	0	0.22231	0	0.05107	0.05107	0	0.48
0.00008	0.57826	0	0.63093	0.19872	0.19262	0.51139	0.57805	0.9257	0.33869	0.88966	0.71311	0.33665	0.60979	0.98915	0.21282	0	0.22231	0	0.05107	0.05107	0	0.48
0.00009	0.57826	0	0.63093	0.19872	0.19262	0.51139	0.57805	0.9257	0.33869	0.88966	0.71311	0.33665	0.60979	0.98915	0.21282	0	0.22231	0	0.05107	0.05107	0	0.48
0.00010	0.57826	0	0.63093	0.19872	0.19262	0.51139	0.57805	0.9257	0.33869	0.88966	0.71311	0.33665	0.60979	0.98915	0.21282	0	0.22231	0	0.05107	0.05107	0	0.48
0.00012	0.57826	0	0.63093	0.19872	0.19262	0.51139	0.57805	0.9257	0.33869	0.88966	0.71311	0.33665	0.60979	0.98915	0.21282	0	0.22231	0	0.05107	0.05107	0	0.48
0.00013	0.57826	0	0.63093	0.19872	0.19262	0.51139	0.57805	0.9257	0.33869	0.88966	0.71311	0.33665	0.60979	0.98915	0.21282	0	0.22231	0	0.05107	0.05107	0	0.48
0.00015	0.57826	0	0.63093	0.19872	0.19262	0.51139	0.57805	0.9257	0.33869	0.88966	0.71311	0.33665	0.60979	0.98915	0.21282	0	0.22231	0	0.05107	0.05107	0	0.48
0.00017	0.57826	0	0.63093	0.19872	0.19262	0.51139	0.57805	0.9257	0.33869	0.88966	0.71311	0.33665	0.60979	0.98915	0.21282	0	0.22231	0	0.05107	0.05107	0	0.48
0.00018	0.57826	0	0.63093	0.19872	0.19262	0.51139	0.57805	0.9257	0.33869	0.88966	0.71311	0.33665	0.60979	0.98915	0.21282	0	0.22231	0	0.05107	0.05107	0	0.48
0.00020	0.57826	0	0.63093	0.19872	0.19262	0.51139	0.57805	0.92571	0.33869	0.88966	0.71311	0.33665	0.60979	0.98915	0.21282	0	0.22231	0	0.05107	0.05107	0	0.48
0.00022	0.57826	0	0.63093	0.19872	0.19262	0.51139	0.57805	0.92571	0.33869	0.88966	0.71311	0.33665	0.60979	0.98915	0.21282	0	0.22231	0	0.05107	0.05107	0	0.48
0.00023	0.57826	0	0.63093	0.19872	0.19262	0.51139	0.57805	0.92571	0.33869	0.88966	0.71311	0.33665	0.60979	0.98915	0.21282	0	0.22231	0	0.05107	0.05107	0	0.48
0.00025	0.57826	0	0.63093	0.19872	0.19262	0.51139	0.57805	0.92571	0.33869	0.88966	0.71311	0.33665	0.60979	0.98915	0.21282	0	0.22231	0	0.05107	0.05107	0	0.48
0.00027	0.57826	0	0.63093	0.19872	0.19262	0.51139	0.57805	0.92571	0.33869	0.88966	0.71311	0.33665	0.60979	0.98915	0.21282	0	0.22231	0	0.05107	0.05107	0	0.48
0.00028	0.57826	0	0.63093	0.19872	0.19262	0.51139	0.57805	0.92571	0.33869	0.88966	0.71311	0.33665	0.60979	0.98915	0.21282	0	0.22231	0	0.05107	0.05107	0	0.48
0.00030	0.57826	0	0.63093	0.19872	0.19262	0.51139	0.57805	0.92571	0.33869	0.88966	0.71311	0.33665	0.60979	0.98915	0.21282	0	0.22231	0	0.05107	0.05107	0	0.48
0.00031	0.57826	0	0.63093	0.19872	0.19262	0.51139	0.57805	0.92571	0.33869	0.88966	0.71311	0.33665	0.60979	0.98915	0.21282	0	0.22231	0	0.05107	0.05107	0	0.48
0.00033	0.57826	0	0.63093	0.19872	0.19262	0.51139	0.57805	0.92571	0.33869	0.88966	0.71311	0.33665	0.60979	0.98915	0.21282	0	0.22231	0	0.05107	0.05107	0	0.48
0.00035	0.57826	0	0.63093	0.19872	0.19262	0.51139	0.57805	0.92571	0.33869	0.88966	0.71311	0.33665	0.60979	0.98915	0.21282	0	0.22231	0	0.05107	0.05107	0	0.48
0.00036	0.57826	0	0.63093	0.19872	0.19262	0.51139	0.57805	0.92571	0.33869	0.88966	0.71311	0.33665	0.60979	0.98915	0.21282	0	0.22231	0	0.05107	0.05107	0	0.48

Figure 5. Snapshot of BoT-IoT dataset after pre-processing stage

4.2.1. Hyper-objective function

The proposed hybrid objective function forms the crux of our approach, evaluating a feature's significance by simultaneously considering its relevance and redundancy. The relevance of a feature is ascertained through its correlation with the target variable, which indicates how much it can contribute to predicting the target. On the other hand, redundancy is measured by the average correlation of the feature with those already selected. The lower the correlation, the less redundant the feature is considered; thus, it brings new, distinct information to the model. The hybrid objective function is calculated using (3):

$$F = (I - C) / (H * V) \quad (3)$$

where I represent the mutual information, indicating the relevance of the feature; C stands for average correlation, used to measure redundancy; H is the entropy of the feature, which reflects the amount of information or 'surprise' the feature offers; and V denotes variance, a measure of how much values of the feature differ, thus capturing its diversity.

Through this, the objective function aims to maximize mutual information and diversity (entropy and variance) while minimizing redundancy, leading to an efficient and more precise feature selection for the IDS.

4.2.2. Mutual information

The correlation between a feature and the target is evaluated using the concept of mutual information (MI). By observing the other, MI quantifies the "amount of information" obtained about one random variable [40]. It's calculated using (4):

$$I(X; Y) = \iint p(x, y) \log \left(\frac{p(x, y)}{p(x)p(y)} \right) dx dy \quad (4)$$

where $p(x, y)$ refers to the combined probability density function of X and Y , $p(x)$ and $p(y)$ are the marginal probability density functions of X and Y , respectively.

In the context of the proposed FS methodology, MI serves as a pivotal metric to ascertain the relevance of a feature by measuring the information gain about the target variable resulting from the feature. The merit of employing MI in this scenario is its non-parametric nature, which enables it to capture non-linear dependencies between variables, thereby enhancing the robustness and comprehensiveness of the feature evaluation process, especially in complex, high-dimensional spaces prevalent in IoT data.

4.2.3. Pearson's correlation coefficient

The pearson correlation coefficient is vital in assessing linear relationships between variables in the proposed hybrid FS methodology. Utilized within the algorithm, it quantifies the degree of linear dependence between selected features and the target variable, thereby offering a measure to evaluate and mitigate redundancy among selected features. Its computation is facilitated through (5):

$$\rho_{X,Y} = \frac{\text{Cov}(X,Y)}{\sigma_X \sigma_Y} \quad (5)$$

let's break down the components of this formula:

$Cov(X, Y)$ is the covariance between X and Y. It measures how much two random variables vary together. It's calculated as the expected value (or mean) of the product of the differences between each variable and their respective means:

$$Cov(X, Y) = E[(X - E[X])(Y - E[Y])] \quad (6)$$

where $(E[X])$ and $(E[Y])$ are the means of X and Y, respectively, and (E) is the expectation or average value. (σ_X) and (σ_Y) are the standard deviations of X and Y. It's calculated as the square root of the variance:

$$\sigma_X = \sqrt{Var(X)} = \sqrt{E[(X - E[X])^2]} \quad (7)$$

$$\sigma_Y = \sqrt{Var(Y)} = \sqrt{E[(Y - E[Y])^2]} \quad (8)$$

where $(Var(X))$ and $(Var(Y))$ are the variances of X and Y, respectively.

4.2.4. Variance

The variance, denoted as $Var(X)$, serves not merely as a statistical measure indicating the deviation of a random variable X from its mean μ , but also as a pivotal factor in assessing the dispersion of feature values within the context of our proposed methodology. It quantitatively portrays how much each feature diverges from its mean, providing insights into its distribution and information dispersion. Integrating variance into the hybrid objective function ensures that features exhibiting a higher degree of dispersion and potentially possessing more informative aspects are assigned elevated fitness scores. This strategy aligns to select features that maximize information gain. The computation of variance adheres to (9):

$$Var(X) = E[(X - \mu)^2] \quad (9)$$

Where $Var(X)$ is signifies the variance of the random variable X. variance serves as a dispersion metric, denoting the degree to which data points deviate from the mean. A heightened variance implies a substantial spread of data points around the mean, while a diminished variance suggests they are closely packed around the mean; E represents the expectation operator, often interpreted as a long-run average value of a random variable. It's also known as the expected value or the mean; X is the random variable. A random variable can be any outcome from some chance process, like your set of possible results from a data group; and μ represents the mean (average) of X. It's calculated by summing all the data points and dividing by the number of data points.

In essence, for each data point, the variance formula takes the difference of the data point from the mean, squares it (to make all differences positive), and then takes an average of these squared differences.

4.2.5. Entropy

Entropy measures the uncertainty, randomness, or impurity in a data set. The entropy (H) of a random variable X is defined as (10):

$$H(X) = - \sum_{i=1}^n p(x_i) \log p(x_i) \quad (10)$$

Where $H(X)$ represents the entropy of the random variable X. In information theory, entropy means the expected amount of 'information' contained in a message. In other words, it measures the unpredictability or uncertainty of a random variable; Σ is the summation symbol. The following expression is added for all values from $i=1$ to n ; $p(x_i)$ is the probability mass function of X. It gives the probability that the random variable X equals some value. In the context of entropy, $p(x_i)$ represents the probability of a particular outcome; and $\log p(x_i)$ is the logarithm of the probability mass function of X. In the context of entropy, taking the logarithm of the probabilities provides a measure of 'information'. The base of the logarithm determines the unit of entropy. If the logarithm is base 2, the entropy is measured in bits.

In the context of the proposed FS methodology, entropy is not merely a statistical measure; it is a critical component ensuring that the selected features embody a spectrum of information about the dataset, thereby enhancing the predictive model's robustness and generalizability. Particularly for high-dimensional IoT data, which often encompasses diverse and non-linear features, the inclusion of entropy in the FS methodology enables the model to discern and prioritize features that encapsulate varied and rich informational content, thereby mitigating the risk of overfitting and enhancing model interpretability.

The innovative methodology outlined for feature selection in IDS seamlessly integrates mutual information, pearson's correlation coefficient, variance, and entropy into a unified hyper-objective function, presenting a comprehensive and balanced approach. This union of various metrics assures a meticulous evaluation and selection of features, maximizing relevance while ensuring diversity and minimizing redundancy. Unlike alternative methodologies such as recursive feature elimination [41], correlation-based feature selection (CFS) [42], linear discriminant analysis (LDA) [43], and principal component analysis (PCA) [44], the proposed method concurrently optimizes multiple critical aspects, enhancing the predictive accuracy and computational efficiency of the IDS. This nuanced approach provides a robust foundation for further research and applications within cybersecurity, demonstrating substantial practical and theoretical value. Algorithm 1 shows a pseudocode of the proposed hybrid objective feature selection methodology for IDS is provided, illustrating the detailed, step-by-step breakdown of the method.

Algorithm 1. Pseudocode of the proposed hybrid objective feature selection methodology for IDS

Input:

X: the dataset of features

y: the target variable

max_features: the maximum number of features to select (default is all features)

fitness_threshold: the threshold below which selection stops (default is None)

Output: selected_features the optimal subset of features

Step 1. **Initialization:** the algorithm begins by initializing two lists. **selected_features** starts as an empty list and will hold the indices of selected features. **remaining_features** is initialized with the index of all features

Step 2. **Function Definition:** hybrid_objective (X, y, feature_idx, selected_features)

Step 3. **Calculate** mutual Information (correlation) between the feature at index **feature_idx** and target variable **y**.

Step 4. **If** **selected_features** is not empty, calculate pairwise correlations between the current and already selected features and take their mean as **avg_correlation**. Otherwise, set **avg_correlation** to 0.

Step 5. **Calculate** entropy **ent** and variance **var** of the feature at index **feature_idx**

Step 6. **Compute** the fitness score as **(correlation - avg_correlation) / (ent * var)**

Step 7. **Return** the fitness score.

Step 8. **Feature Selection:**

Step 9. **For** **i = 0 to max_features do:**

Step 10. **Initialize** **best_fitness** as negative infinity and **best_feature** as None.

Step 11. **For each** **feature_idx** **in** **remaining_features** **do:**

Step 12. **Compute** the fitness score using **hybrid_objective(X, y, feature_idx, selected_features)**

Step 13. **If** the fitness score is more significant than **best_fitness**, **update** **best_fitness** and **best_feature**

Step 14. **Append** **best_feature** to **selected_features** and remove it from **remaining_features**

Step 15. **Break the loop if fitness_threshold is provided and best_fitness is less than fitness_threshold.**

Step 16. **Return selected_features**

- X: the dataset of features that represent the input data.
- y: the target variable is the variable being classified.
- max_features: the maximum number of features to select. The user can define this parameter to limit the number of selected features. If it's not provided, the algorithm will consider all the features from the dataset.
- fitness_threshold: this is the threshold which the selection process stops. It's another way of controlling the stopping condition of the FS process. If the calculated fitness score falls below this threshold, the algorithm will stop adding more features to the selected subset. If this parameter is not provided, the algorithm will only stop when it has considered all features or when the maximum number of selected features has been reached.
- Initialization: the algorithm begins by initializing two lists. selected_features start as an empty list and will hold the indices of selected features. remaining_features is initialized with the indices of all features.
- Function definition: the hybrid_objective function is crucial to the algorithm. It computes a fitness score for a given feature, determining its relevance and suitability for inclusion in the selected feature subset.

The feature fitness score is computed as the difference between the mutual information and the average correlation, divided by the product of the entropy and the variance according to (3).

The intuition behind this fitness score is to maximize the mutual information with the target variable, minimize the redundancy with already selected features, and favor features with high entropy and variance.

- Feature selection: the main iteration of the algorithm begins and continues until all features have been considered or the maximum number of selected features (max_features) has been reached.

In each iteration of this loop, the algorithm considers each remaining feature and calculates its fitness score using the hybrid_objective function. If the fitness score of a feature is higher than the best fitness score so far, it becomes the new best feature, and the best fitness score is updated. This process continues for all remaining features, thus ensuring that the feature with the highest fitness score in each round is selected.

After each cycle, the chosen attribute is incorporated into the list of selected features and subtracted from the remaining features. This principle safeguards against the repetition of the same attribute. If a fitness threshold parameter is established and the peak fitness score dips beneath this marker, the algorithm will prematurely cease, irrespective of the total count of selected features.

- Output: after the initial run of the algorithm, we receive a collection of selected features. This collection can generate a simplified version of the original dataset, including only the most relevant features. These selected features can be used later to build a classification model.

The hybrid fitness function used in the forward selection method is a robust and adaptable algorithm. Its primary strength resides in its capacity to evaluate the significance of each feature relative to the target variable using mutual information, as well as the redundancy of each feature concerning those already selected using correlation. Furthermore, this strategy considers each feature's entropy and variance. Considering various aspects of the data enables the formation of a thorough and insightful set of features that can result in high performance in following predictive modeling tasks.

Distinctively, this algorithm integrates the strengths of various existing methods and mitigates their limitations by providing a balanced, well-rounded evaluative mechanism, thereby amplifying its reliability and applicability in real-world scenarios. It substantiates a novel paradigm in feature selection by concurrently maximizing mutual information and diversity while minimizing redundancy, thereby ensuring a meticulous and nuanced selection of features that not only are relevant but also enhance the predictive model's robustness and generalizability.

4.2.6. Classifier (BiLSTM)

The classifier utilized in this study is a BiLSTM network, a variant of RNNs. BiLSTM models have gained considerable prominence in various domains due to their ability to capture long-term dependencies and temporal patterns in sequential data. The architecture of the BiLSTM network incorporates LSTM layers that operate in both forward and backward directions [45]. This dual-directional operation is vividly illustrated in Figure 6. The forward LSTM mechanism examines the input sequence progressively from the beginning to the end, capturing data from historical events.

On the other hand, the backward LSTM mechanism reviews the sequence in reverse order, acquiring knowledge from upcoming events [46]. This two-way processing is depicted in Figure 4. The amalgamation of outputs from these two directions enables the BiLSTM model to scrutinize the context of the input sequence comprehensively. This results in successfully detecting and extracting relationships and patterns over various time scales [47].

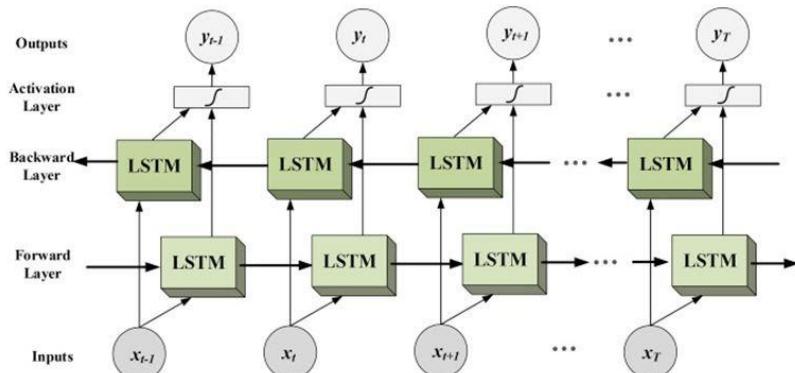


Figure 6. BiLSTM architecture

5. EXPERIMENTAL SETUP

The experimental design, a core part of the research methodology, sets up the framework for evaluating the effectiveness and reliability of the proposed method in classifying IoT data. Key components include dataset selection, model configurations, evaluation metrics, and verification methods. Datasets TON_IoT and BoT-IoT, which contain a variety of IoT activities, are used. The BiLSTM classifier's parameters and hyperparameters are defined to enhance learning efficiency. The evaluation uses metrics like accuracy, precision, recall, and F1-score. Validation techniques like train-test splits prevent overfitting and test the model's generalizability. Adherence to best practices in experimental design and statistical analysis ensures a robust framework for assessing the proposed IoT data classification approach.

5.1. Datasets

In this study, the BoT-IoT [48] and TON_IoT [49] datasets are utilized, capturing a diverse range of IoT activities and offering valuable insights for evaluating the proposed approach in intrusion detection.

5.1.1. BoT-IoT dataset

The BoT-IoT dataset was created by utilizing industrial IoT (IIoT) smart home appliances in collecting IIoT traffic samples within the Cyber Range Lab of The Center of UNSW Canberra Cyber. This dataset encompasses many smart IIoT devices, such as thermostats, motion-controlled lights, remotely controlled garages, fridges and freezers, and weather monitoring systems. Two versions of the dataset are available: the full version, consisting of over 72 million records, and the 10% version, which includes approximately 3.6 million records. For our experimentation, a subset of the dataset, specifically 5% of the entire dataset, has been chosen for analysis. The focus is on the top ten features that demonstrate the best performance.

5.1.2. TON_IoT dataset

The TON-IoT dataset used in this study is a comprehensive collection of heterogeneous data from a medium-scale IoT network. It includes telemetry data, operating system records, and network traffic data. The dataset is labeled, indicating normal behavior or attacks such as ransomware, password attacks, DoS, and DDoS. The dataset was created in collaboration between UNSW Canberra IoT Labs and the Cyber Range and can be accessed in CSV format [50].

5.2. Model training

The model training involves defining the architecture of the BiLSTM deep learning model, compiling the model with appropriate parameters, and training the model using the training dataset. For a detailed overview of the hyperparameters and configuration used in this process, refer to Table 1. This table outlines critical parameters such as the number of epochs, batch size, learning rate, and loss function, which are crucial for the training and performance of the BiLSTM model.

Table 1. Hyperparameters and configuration for BiLSTM model training

Parameter	Value
Epochs	100
Batch size	32
Learning rate	0.001
Loss function	Binary cross entropy
Optimizer	Adam

5.3. Evaluation metrics

The evaluation metrics play a crucial role in assessing the performance of the classification model in this work. Standard evaluation metrics, such as accuracy, precision, recall, and F1-score, provide quantitative measures of the model's effectiveness in classifying normal and malicious IoT activities. These metrics can be represented using (11) to (14):

$$\text{Accuracy} = \frac{\text{Number of correctly classified instances}}{\text{Total number of instances}} \quad (11)$$

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \quad (12)$$

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \quad (13)$$

$$F1\text{-score} = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (14)$$

6. RESULTS

The presented methodology for constructing an efficient (IDS) in fog computing demonstrates its proficiency through our study's results. Evaluation metrics quantitatively express this proficiency, including accuracy, precision, recall, and F1-score.

6.1. Feature selection

The proposed hybrid objective FS algorithm significantly reduced the dimensionality of the initial datasets. As depicted in Figures 7 and 8, the features declined substantially through the TON_IoT and BoT-IoT datasets selection process.

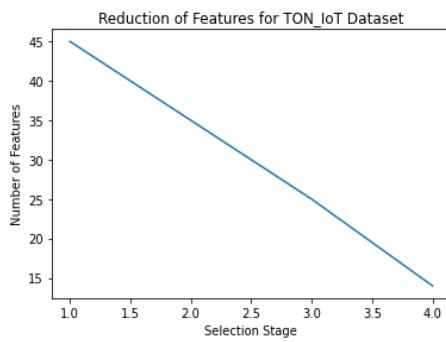


Figure 7. Reduction of features for the TON_IoT dataset

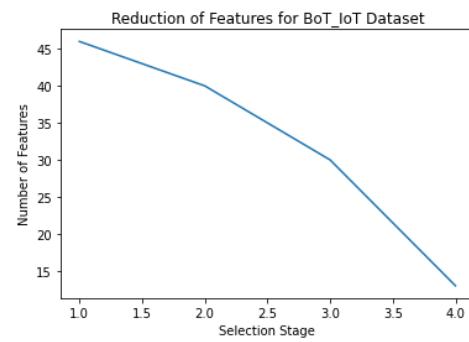


Figure 8. Reduction of features for the BoT_IoT dataset

The process began with a comprehensive set of features, and the list was methodically culled to include only those with a fitness score less than the defined threshold of 0.2. This refined FS contributes to a more efficient and manageable workflow, with the most pertinent data being preserved for further steps. The details of feature reduction in the TON_IoT and BoT-IoT datasets are shown in Table 2.

Table 2. Feature selection in TON_IoT and BoT-IoT datasets

Dataset	Initial number of features	Number of features after selection
TON_IoT	45	14
BoT-IoT	46	13

The tables indicate that the initial number of features was significantly reduced after applying our hybrid objective FS algorithm. For instance, in the TON_IoT dataset, we started with 45 features and narrowed the list to only 14 significant features. Similarly, in the BoT-IoT dataset, the initial feature set containing 46 features was reduced to 13. This reduction not only streamlined the datasets but also improved the efficiency of the following stages in the workflow, including training the BiLSTM classifier.

6.2. Performance on TON_IoT and BoT-IoT dataset

In Table 3, the BiLSTM model achieved high performance on the TON_IoT and BoT-IoT datasets. The accuracy for TON_IoT was 98.42%, and BoT-IoT's was 98.7%. The precision values were 98.3% for TON_IoT and 98.1% for BoT-IoT. The model exhibited a recall of 97.7% for both datasets. Additionally, the F1 scores for both datasets were 98%. These performance metrics indicate the model's accuracy, precision, recall, and balanced performance in predicting activities for both TON_IoT and BoT-IoT datasets.

Table 3. Performance metrics of the BiLSTM model on TON_IoT and BoT-IoT datasets

Metric	Value (%) (TON_IoT)	Value (%) (BoT-IoT)
Accuracy	98.42	98.7
Precision	98.3	98.1
Recall	97.7	97.7
F1-score	98	98

6.3. Overall performance

The combination of the hybrid FS method and the BILSTM classifier exhibited remarkable efficiency in discerning normal from potentially malicious IoT activities. The high performance on both TON_IoT and BoT-IoT datasets significantly advances IoT security. These findings suggest that ML and DL techniques, such as BILSTM models, hold great potential for constructing robust IDS within fog computing.

Figures 9 and 10 illustrate the trajectory of training and validation losses during the model training process. It is an essential tool for monitoring model performance, allowing the identification of possible overfitting or underfitting situations by comparing the behavior of loss on both the training and validation datasets over the training epochs.

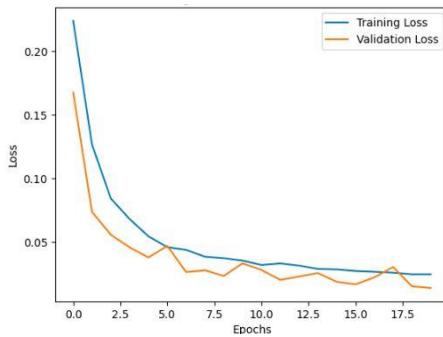


Figure 9. Training and validation loss curve for TON_IoT

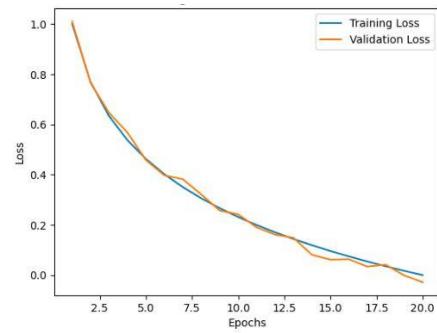


Figure 10. Training and validation loss curve for BoT-IoT

Figures 11 and 12 display the model's accuracy on the training and validation datasets over the training epochs. This allows for tracking the model's learning progress and assessing whether the model might be overfitting or underfitting.

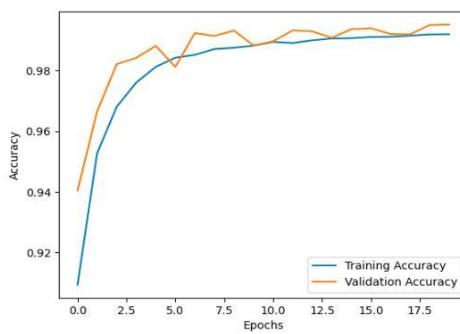


Figure 11. Training and validation accuracy curve for TON_IoT

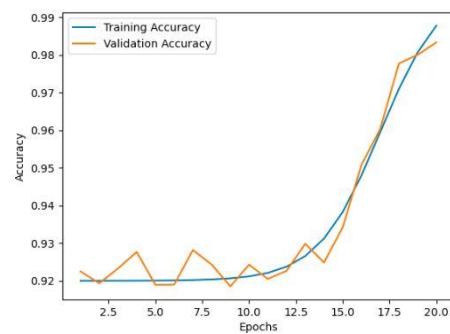


Figure 12. Training and validation accuracy curve for BoT-IoT

The hybrid objective function in the study is an integral part of the algorithm, computing a fitness score for each feature to ascertain its relevance and suitability for inclusion in the selected feature subset. The function calculates the feature fitness score using an equation that measures the difference between the mutual information and the average correlation, divided by the product of entropy and variance. The underlying intuition of this fitness score is to maximize the mutual information with the target variable, thereby ensuring that the feature contributes meaningful information for classification. It also aims to minimize redundancy with already selected features, ensuring that each selected feature brings new, distinct information to the model. Lastly, by favoring features with high entropy and variance, the algorithm prioritizes features with a high degree of information richness and diversity.

This objective function plays a crucial role in improving the performance of the IDS. Accurately gauging the importance of each feature ensures that the model learns from the most pertinent and informative features, leading to a more accurate and efficient intrusion detection system. The performance of the

BiLSTM IDS is benchmarked against several premier IDSs using the evaluation metrics specified in Section 5.3. This comparative assessment aids in determining the accuracy, precision, recall, and F-measure of the BiLSTM IDS for identifying attacks within a fog environment compared to similar IDSs. Included in the comparison are IDSs like the LSTM-IDS [51], CNN-IDS [52], RNN-IDS [53], and GRN-RNN IDS [54], chosen as benchmark models due to their analogous performance levels. Table 4 showcases the evaluation metrics for the BiLSTM IDS and these leading-edge IDSs. The data from Table 4 highlights the BiLSTM IDS's supremacy over all the examined IDSs in accuracy, precision, recall, and F-measure. This exceptional performance is credited to the system's effective hyper objectives feature selection model, which markedly improves its ability to identify attacks in the fog environment.

Table 4. Comparison with state-of-the-art IDSs

Metric	Proposed	LSTM-IDS	CNN-IDS	RNN-IDS	GRN-RNN IDS
Accuracy	98.7	97.7	95.5	91.7	92
Precision	98.1	97	96	99	99
Recall	97.7	95	97	90.2	92.05
F1-score	98	96	97	94.6	95.75

7. CONCLUSION

The effectiveness of the proposed BiLSTM model in IoT data classification tasks was validated by our experimental results, with high performance demonstrated on both the BoT-IoT and TON_IoT datasets. In our experiments, the BiLSTM model achieved an accuracy of 98.42% on the TON_IoT dataset and 98.7% on the BoT-IoT dataset. The precision was recorded at 98.3% for TON_IoT and 98.1% for BoT-IoT, while recall was consistent at 97.7% for both datasets. The F1 score stood at 98% across both datasets. This robustness suggests the model's capability across diverse IoT activities, displaying remarkable precision in identifying malicious activities and high recall rates, proving its real-world applicability. In feature selection, the benefits of using hybrid objective/fitness functions such as correlation, entropy, and variance stand out. By combining multiple statistical measures, these hybrid methods offer a comprehensive assessment of feature relevance, capturing intricate relationships between features and leading to improved model performance, reduced overfitting, and a more interpretable feature set.

However, the performance of deep learning models, like the implemented BiLSTM, can be influenced by various factors, including hyperparameters, architecture complexity, and dataset characteristics. Further research and fine-tuning might be essential for optimal performance across contexts. While we acknowledge the importance of real-time detection, especially in fog computing, this study focused on offline training and testing modes. The real-time aspect, crucial for actual environments, remains outside this work's scope but is a focal point for our future research endeavors.

In summary, this research underscores the potential of deep learning methodologies, specifically BiLSTM, in complex classification tasks within the IoT and fog computing domains, paving the way for future advancements in data classification and intrusion detection.

ACKNOWLEDGEMENTS

This research is funded by Universiti Sains Malaysia (USM) via an external grant (number. 304/PNAV/650958/U154).

REFERENCES

- [1] A. Thakkar and R. Lohiya, "A survey on intrusion detection system: feature selection, model, performance measures, application perspective, challenges, and future research directions," *Artificial Intelligence Review*, vol. 55, no. 1, pp. 453–563, Jan. 2022, doi: 10.1007/s10462-021-10037-9.
- [2] Y. Harbi, Z. Aliouat, and S. Harous, "Fog Computing Security and Privacy for Internet of Things (IoT) and Industrial Internet of Things (IIoT) Applications: State of the Art," in *Internet of Things*, 2022, pp. 145–157, doi: 10.1007/978-3-031-08254-2_9.
- [3] D. Mohamed and O. Ismael, "Enhancement of an IoT hybrid intrusion detection system based on fog-to-cloud computing," *Journal of Cloud Computing*, vol. 12, no. 1, pp. 1–13, 2023, doi: 10.1186/s13677-023-00420-y.
- [4] Y. K. Saheed, A. I. Abiodun, S. Misra, M. K. Holone, and R. Colomo-Palacios, "A machine learning-based intrusion detection for detecting internet of things network attacks," *Alexandria Engineering Journal*, vol. 61, no. 12, pp. 9395–9409, 2022, doi: 10.1016/j.aej.2022.02.063.
- [5] J. Cai, J. Luo, S. Wang, and S. Yang, "Feature selection in machine learning: A new perspective," *Neurocomputing*, vol. 300, pp. 70–79, 2018, doi: 10.1016/j.neucom.2017.11.077.
- [6] M. H. Kamarudin, C. Maple, and T. Watson, "Hybrid feature selection technique for intrusion detection system," *International Journal of High Performance Computing and Networking*, vol. 13, no. 2, p. 232, 2019.
- [7] Y. Yang, S. Tu, R. H. Ali, H. Alasmary, M. Waqas, and M. N. Amjad, "Intrusion Detection Based on Bidirectional Long Short-

Term Memory with Attention Mechanism," *Computers, Materials and Continua*, vol. 74, no. 1, pp. 801–815, 2023, doi: 10.32604/cmc.2023.031907.

[8] N. Koroniotis, N. Moustafa, E. Sitnikova, and J. Slay, "towards developing network forensic mechanism for botnet activities in the IoT based on machine learning techniques," *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, LNICST*, vol. 235, pp. 30–44, 2018, doi: 10.1007/978-3-319-90775-8_3.

[9] F. A. Zwayed, M. Anbar, Y. Sanjalawe, and S. Manickam, "Intrusion Detection Systems in Fog Computing – A Review," *Communications in Computer and Information Science (CCIS)*, vol. 1487, pp. 481–504, 2021, doi: 10.1007/978-981-16-8059-5_30.

[10] M. H. Kashani, A. M. Rahmani, and N. J. Navimipour, "Quality of service-aware approaches in fog computing," *International Journal of Communication Systems*, vol. 33, no. 8, 2020, doi: 10.1002/dac.4340.

[11] Z. R. Alashhab, M. Anbar, M. M. Singh, Y. B. Leau, Z. A. Al-Sai, and S. A. Alhayja'a, "Impact of Coronavirus Pandemic Crisis on Technologies and Cloud Computing Applications," *Journal of Electronic Science and Technology*, vol. 19, no. 1, pp. 25–40, Mar. 2021, doi: 10.1016/j.jnest.2020.100059.

[12] M. R. Anawar, S. Wang, M. A. Zia, A. K. Jadoon, U. Akram, and S. Raza, "Fog Computing: An Overview of Big IoT Data Analytics," *Wireless Communications and Mobile Computing*, vol. 2018, 2018, doi: 10.1155/2018/7157192.

[13] M. Muniswamaia, T. Agerwala, and C. C. Tappert, "Fog Computing and the Internet of Things (IoT): A Review," *Proceedings - 2021 8th IEEE International Conference on Cyber Security and Cloud Computing and 2021 7th IEEE International Conference on Edge Computing and Scalable Cloud, CSCloud-EdgeCom 2021*, vol. 2, no. 2, pp. 10–12, 2021, doi: 10.1109/CSCloud-EdgeCom52276.2021.00012.

[14] S. Khan, S. Parkinson, and Y. Qin, "Fog computing security: a review of current applications and security solutions," *Journal of Cloud Computing*, vol. 6, no. 1, p. 19, Dec. 2017, doi: 10.1186/s13677-017-0090-3.

[15] K. Alieyan, M. M. Kadhum, M. Anbar, S. U. Rehman, and N. K. A. Alajmi, "An overview of DDoS attacks based on DNS," in *2016 International Conference on Information and Communication Technology Convergence, ICTC 2016*, IEEE, Oct. 2016, pp. 276–280, doi: 10.1109/ICTC.2016.7763485.

[16] M. Mukherjee *et al.*, "Security and Privacy in Fog Computing: Challenges," *IEEE Access*, vol. 5, pp. 19293–19304, 2017, doi: 10.1109/ACCESS.2017.2749422.

[17] Y. Rhab *et al.*, "Machine Learning and Deep Learning Methods for Intrusion Detection Systems in IoMT: A survey," *2022 2nd International Conference on Innovative Research in Applied Science, Engineering and Technology, IRASET 2022*, vol. 4396, no. 9, 2022, doi: 10.1109/IRASET52964.2022.9738218.

[18] A. Samy, H. Yu, and H. Zhang, "Fog-Based Attack Detection Framework for Internet of Things Using Deep Learning," *IEEE Access*, vol. 8, pp. 74571–74585, 2020, doi: 10.1109/ACCESS.2020.2988854.

[19] N. Al-Mi'ani, M. Anbar, Y. Sanjalawe, and S. Karuppayah, "Securing Software Defined Networking Using Intrusion Detection System - A Review," in *Communications in Computer and Information Science*, 2021, pp. 417–446, doi: 10.1007/978-981-16-8059-5_26.

[20] A. K. Al-Ani, M. Anbar, A. Al-Ani, and D. R. Ibrahim, "Match-Prevention Technique against Denial-of-Service Attack on Address Resolution and Duplicate Address Detection Processes in IPv6 Link-Local Network," *IEEE Access*, vol. 8, pp. 27122–27138, 2020, doi: 10.1109/ACCESS.2020.2970787.

[21] B. B. Zarpelão, R. S. Miani, C. T. Kawakami, and S. C. de Alvarenga, "A survey of intrusion detection in Internet of Things," *Journal of Network and Computer Applications*, vol. 84, pp. 25–37, 2017, doi: 10.1016/j.jnca.2017.02.009.

[22] Y. Saeks, I. Inza, and P. Larrañaga, "A review of feature selection techniques in bioinformatics," *Bioinformatics*, vol. 23, no. 19, pp. 2507–2517, 2007, doi: 10.1093/bioinformatics/btm344.

[23] G. C. and F. Sahin, "A survey on feature selection methods," *Computers & Electrical Engineering*, vol. 40, no. 1, pp. 16–28, 2014.

[24] R. Kohavi and G. H. John, "Wrappers for feature subset selection," *Artificial Intelligence*, vol. 97, no. 1–2, pp. 273–324, 1997, doi: 10.1016/s0004-3702(97)00043-x.

[25] R. Singh and R. L. Ujjwal, "Feature Selection Methods for IoT Intrusion Detection System: Comparative Study," *Lecture Notes in Electrical Engineering*, vol. 968, pp. 227–236, 2023, doi: 10.1007/978-981-19-7346-8_20.

[26] B. Venkatesh and J. Anuradha, "A review of Feature Selection and its methods," *Cybernetics and Information Technologies*, vol. 19, no. 1, pp. 3–26, 2019, doi: 10.2478/CAIT-2019-0001.

[27] F. Kamalov, S. Moussa, R. Zgheib, and O. Mashaal, "Feature selection for intrusion detection systems," *Proceedings - 2020 13th International Symposium on Computational Intelligence and Design, ISCID 2020*, pp. 265–269, 2020, doi: 10.1109/ISCID51228.2020.00065.

[28] S. Lei, C. Xia, Z. Li, X. Li, and T. Wang, "HNN: A Novel Model to Study the Intrusion Detection Based on Multi-Feature Correlation and Temporal-Spatial Analysis," *IEEE Transactions on Network Science and Engineering*, vol. 8, no. 4, pp. 3257–3274, 2021, doi: 10.1109/TNSE.2021.3109644.

[29] A. A. Salih and M. B. Abdulrazaq, "Combining Best Features Selection Using Three Classifiers in Intrusion Detection System," *2019 International Conference on Advanced Science and Engineering, ICOASE 2019*, pp. 94–99, 2019, doi: 10.1109/ICOASE.2019.8723671.

[30] J. O. Onah, S. M. Abdulhamid, M. Abdullahi, I. H. Hassan, and A. Al-Ghusham, "Genetic Algorithm based feature selection and Naïve Bayes for anomaly detection in fog computing environment," *Machine Learning with Applications*, vol. 6, p. 100156, 2021, doi: 10.1016/j.mlwa.2021.100156.

[31] K. Albulayhi, Q. A. Al-Haija, S. A. Alsuhibany, A. A. Jillepalli, M. Ashrafuzzaman, and F. T. Sheldon, "IoT Intrusion Detection Using Machine Learning with a Novel High Performing Feature Selection Method," *Applied Sciences (Switzerland)*, vol. 12, no. 10, p. 5015, May 2022, doi: 10.3390/app12105015.

[32] M. B. Pranto, M. H. A. Ratul, M. M. Rahman, I. J. Diya, and Z. Bin Zahir, "Performance of Machine Learning Techniques in Anomaly Detection with Basic Feature Selection Strategy-A Network Intrusion Detection System," *Journal of Advances in Information Technology*, vol. 13, no. 1, pp. 36–44, 2022, doi: 10.12720/jait.13.1.36-44.

[33] A. Thakkar and R. Lohiya, "Fusion of statistical importance for feature selection in Deep Neural Network-based Intrusion Detection System," *Information Fusion*, vol. 90, pp. 353–363, 2023, doi: 10.1016/j.inffus.2022.09.026.

[34] J. G. Almaraz-Rivera, J. A. Perez-Diaz, and J. A. Cantoral-Ceballos, "Transport and Application Layer DDoS Attacks Detection to IoT Devices by Using Machine Learning and Deep Learning Models," *Sensors*, vol. 22, no. 9, p. 3367, Apr. 2022, doi: 10.3390/s22093367.

[35] G. Zhao, Y. Wang, and J. Wang, "Lightweight Intrusion Detection Model of the Internet of Things with Hybrid Cloud-Fog Computing," *Security and Communication Networks*, vol. 2023, pp. 1–16, Jan. 2023, doi: 10.1155/2023/7107663.

[36] A. T. Nururrahmah and T. Ahmad, "Feature Selection for Intrusion Detection Using Independence Level Test with an Exhaustive Approach," *International Journal of Intelligent Engineering and Systems*, vol. 16, no. 5, pp. 637–648, Oct. 2023, doi: 10.22266/ijies2023.1031.54.

[37] T. Hamed, J. B. Ernst, and S. C. Kremer, "A survey and taxonomy on data and pre-processing techniques of intrusion detection systems," *Computer and Network Security Essentials*, pp. 113–134, 2017, doi: 10.1007/978-3-319-58424-9_7.

[38] T. Ahmad and M. N. Aziz, "Data preprocessing and feature selection for machine learning intrusion detection systems," *ICIC Express Letters*, vol. 13, no. 2, pp. 93–101, 2019, doi: 10.24507/icicel.13.02.93.

[39] C. Ribeiro and A. A. Freitas, "A data-driven missing value imputation approach for longitudinal datasets," *Artificial Intelligence Review*, vol. 54, no. 8, pp. 6277–6307, 2021, doi: 10.1007/s10462-021-09963-5.

[40] M. Beraha, A. M. Metelli, M. Papini, A. Tirinzoni, and M. Restelli, "Feature Selection via Mutual Information: New Theoretical Insights," *Proceedings of the International Joint Conference on Neural Networks*, vol. 2019, 2019, doi: 10.1109/IJCNN.2019.8852410.

[41] M. Awad and S. Fraihat, "Recursive Feature Elimination with Cross-Validation with Decision Tree: Feature Selection Method for Machine Learning-Based Intrusion Detection Systems," *Journal of Sensor and Actuator Networks*, vol. 12, no. 5, p. 67, Sep. 2023, doi: 10.3390/jsan12050067.

[42] E. S. Alomari *et al.*, "Malware Detection Using Deep Learning and Correlation-Based Feature Selection," *Symmetry*, vol. 15, no. 1, p. 123, Jan. 2023, doi: 10.3390/sym15010123.

[43] A. F. H. Alharan, Z. M. Algelal, N. S. Ali, and N. Al-Garaawi, "Improving Classification Performance for Diabetes with Linear Discriminant Analysis and Genetic Algorithm," in *Proceedings - 2021 Palestinian International Conference on Information and Communication Technology, PICICT 2021*, IEEE, Sep. 2021, pp. 38–44, doi: 10.1109/PICICT53635.2021.00019.

[44] M. A. Almaiah *et al.*, "Performance Investigation of Principal Component Analysis for Intrusion Detection System Using Different Support Vector Machine Kernels," *Electronics (Switzerland)*, vol. 11, no. 21, p. 3571, Nov. 2022, doi: 10.3390/electronics11213571.

[45] S. Siami-Namini, N. Tavakoli, and A. S. Namin, "The Performance of LSTM and BiLSTM in Forecasting Time Series," *Proceedings - 2019 IEEE International Conference on Big Data, Big Data 2019*, pp. 3285–3292, 2019, doi: 10.1109/BigData47090.2019.9005997.

[46] A. Diro and N. Chilamkurti, "Leveraging LSTM Networks for Attack Detection in Fog-to-Things Communications," *IEEE Communications Magazine*, vol. 56, no. 9, pp. 124–130, Sep. 2018, doi: 10.1109/MCOM.2018.1701270.

[47] Y. Dai, Z. Wu, and H. Zhang, "Sentiment Analysis of Comment Texts Based on CNN-BiGRU-Attention," *Proceeding - 2021 China Automation Congress, CAC 2021*, vol. 7, pp. 2749–2754, 2021, doi: 10.1109/CAC53003.2021.9728140.

[48] N. Koroniots, N. Moustafa, E. Sitnikova, and B. Turnbull, "Towards the development of realistic botnet dataset in the Internet of Things for network forensic analytics: Bot-IoT dataset," *Future Generation Computer Systems*, vol. 100, pp. 779–796, 2019, doi: 10.1016/j.future.2019.05.041.

[49] N. Moustafa, "A new distributed architecture for evaluating AI-based security systems at the edge: Network TON_IoT datasets," *Sustainable Cities and Society*, vol. 72, p. 102994, Sep. 2021, doi: 10.1016/j.scs.2021.102994.

[50] A. Alsaedi, N. Moustafa, Z. Tari, A. Mahmood, and Adna N Anwar, "TON-IoT telemetry dataset: A new generation dataset of IoT and IIoT for data-driven intrusion detection systems," *IEEE Access*, vol. 8, pp. 165130–165150, 2020, doi: 10.1109/ACCESS.2020.3022862.

[51] R. Chaganti, W. Suliman, V. Ravi, and A. Dua, "Deep Learning Approach for SDN-Enabled Intrusion Detection System in IoT Networks," *Information (Switzerland)*, vol. 14, no. 1, p. 41, Jan. 2023, doi: 10.3390/info14010041.

[52] A. Henry *et al.*, "Composition of Hybrid Deep Learning Model and Feature Optimization for Intrusion Detection System," *Sensors*, vol. 23, no. 2, p. 890, Jan. 2023, doi: 10.3390/s23020890.

[53] A. Meliboev, J. Alikhanov, and W. Kim, "Performance Evaluation of Deep Learning Based Network Intrusion Detection System across Multiple Balanced and Imbalanced Datasets," *Electronics (Switzerland)*, vol. 11, no. 4, p. 515, Feb. 2022, doi: 10.3390/electronics11040515.

[54] W. Elmasry, A. Akbulut, and A. H. Zaim, "Evolving deep learning architectures for network intrusion detection using a double PSO metaheuristic," *Computer Networks*, vol. 168, p. 107042, Feb. 2020, doi: 10.1016/j.comnet.2019.107042.

BIOGRAPHIES OF AUTHORS



Fadi Abu Zwayed is a Ph.D. Student at the National Advanced IPv6 Centre of Excellence (NAv6), University Sains Malaysia. He received a B.Sc. in Computer Science from Alhussien Bin Talal University, Jordan, in 2007 and an M.Sc. in Computer Science from Al-Balqa' Applied University, Jordan, in 2014. His research interests are in the areas of cybersecurity, cloud computing, and machine learning. He can be contacted at email: f.abuzwayed@gmail.com.



Mohammed Anbar obtained his Ph.D. degree in advanced computer network from USM in 2013. He is currently a senior lecturer with NAv6, USM. His current research interests include malware detection, web security, intrusion detection systems (IDS), intrusion prevention systems (IPS), network monitoring, IoT, and internet protocol version 6 (IPv6) security. He can be contacted at email: anbar@usm.my.



Selvakumar Manickam     is currently working as an associate professor at National Advanced IPv6 Centre (NAv6), Universiti Sains Malaysia. His research interests include cybersecurity, the internet of things, industry 4.0, and machine learning. He has authored and co-authored more than 160 articles in journals, conference proceedings, and book reviews and graduated 13 Ph.D. He had ten years of industrial experience before joining academia. He is a member of technical forums at national and international levels. He also has experience building IoT, embedded, server, mobile, and web-based applications. He can be contacted at email: selva@usm.my.



Yousef Sanjalawe     holds a Ph.D. degree in Cloud Computing-Cybersecurity from Universiti Sains Malaysia (USM) in 2020, Penang, Malaysia. He is currently an Assistant Professor at the Department of Cybersecurity, School of Information Technology at the American University of Madaba (AUM). He served as a field supervisor for Ph.D. students in different fields, including cybersecurity, cloud computing, IoT, fog computing, optimization, and AI. His main research interests are AI, cybersecurity, optimization, cloud computing, and IoT. He can be contacted at email: Yousefsinjlawi@gmail.com.



Hamza Alrababah     received his B.Sc. degree in 2008, M.Sc. in 2010, and Ph.D. in 2015 from Donetsk National Technical University, Ukraine. His major is computer science, and he specializes in software engineering. He has worked at Al Jazeerah University in Dubai, Sharjah University, and Al Wasl University in Dubai. Currently he is teaching in the school of computing at Skyline University College in Sharjah-UAE. His research interests include software engineering, AI, biomedical engineering, network engineering, and cybersecurity. He can be contacted at email: hamza.alrababah@skylineuniversity.ac.ae.



Iznan H. Hasbullah     holds a B.Sc. degree in Electrical Engineering from Rensselaer Polytechnic Institute, Troy, NY, USA, and an M.Sc. degree in advanced Internet security and monitoring from Universiti Sains Malaysia (USM). He has experience as a software developer, research and development consultant, CTO, and network security auditor before joining the National Advanced IPv6 Centre (NAv6) in 2010 as a Research Officer. His research interests include unified communication, network and Internet security, computer network protocols, and next-generation networks. He can be contacted at email: iznan@usm.my.



Noor Almi'ani     is a Ph.D Student at the National Advanced IPv6 Centre of Excellence (NAv6), University Sains Malaysia. She received a B.Sc. in Software Engineering from Alhussien Bin Talal University, Jordan, in 2013 and an M.Sc. in Information System Security and Digital Criminology from Princess Sumaya University for Technology, Jordan, in 2017. Her research interests are in the areas of cybersecurity, software-defined networking, and machine learning. She can be contacted at email: Nyswe1991@gmail.com.